

# **OPERATING SYSTEMS**

## **SECURITY**

# SECURITY

## In This Chapter:

The Security Problem

- Program Threats
- System and Network Threats
- Cryptography as a Security Tool
- User Authentication
- Implementing Security Defenses
- Firewalling to Protect Systems and Networks
- Computer-Security Classifications
- An Example: Windows XP

# SECURITY

## SECURITY ISSUES:

**External** protection of a system. A classified site goes to extraordinary lengths to keep things physically tight. Among the issues to be considered:

<b>Unauthorized access</b>	Mechanism assuring only authorized individuals see classified materials.
<b>Malicious</b>	modification or destruction
<b>Accidental</b>	introduction of inconsistency.
<b>Authentication</b>	Can have passwords on domains.

**Protection of passwords** is difficult. Issues include:

- guess passwords since people use simple and easily remembered words.
- Need exists to change passwords continually.
- Limiting number of tries before locking up.

# SECURITY

<b>Trojan Horse:</b>	A code that misuses its environment. The program seems innocent enough, when executed, unexpected behavior occurs.
<b>Trap Doors:</b>	some secret set of inputs to a program might provide special privileges.
<b>Threat monitoring:</b>	Look for unusual activity. identify someone acting in an unusual fashion?
<b>Audit Log:</b>	Record time, user, and type of access on all objects. Trace problems back to source.
<b>Internet Worm:</b>	In the Internet worm, Robert Morse exploited UNIX networking features (remote access) as well as bugs in send mail programs. Grappling hook program uploaded main worm program.
<b>Viruses</b>	Fragment of code embedded in a program. Mainly effects personal PC systems. These are often downloaded via e-mail or active web pages.
<b>Firewall</b>	A mechanism that allows only certain traffic between trusted and un-trusted systems. Often applied to a way to keep unwanted internet traffic away from a system.

# Typical Security Attacks

## ATTACK METHODS:

Passive wiretapping. ( unauthorized interception /reading of messages )

- Active wiretapping:

**Modification**      Changing a portion of the message.

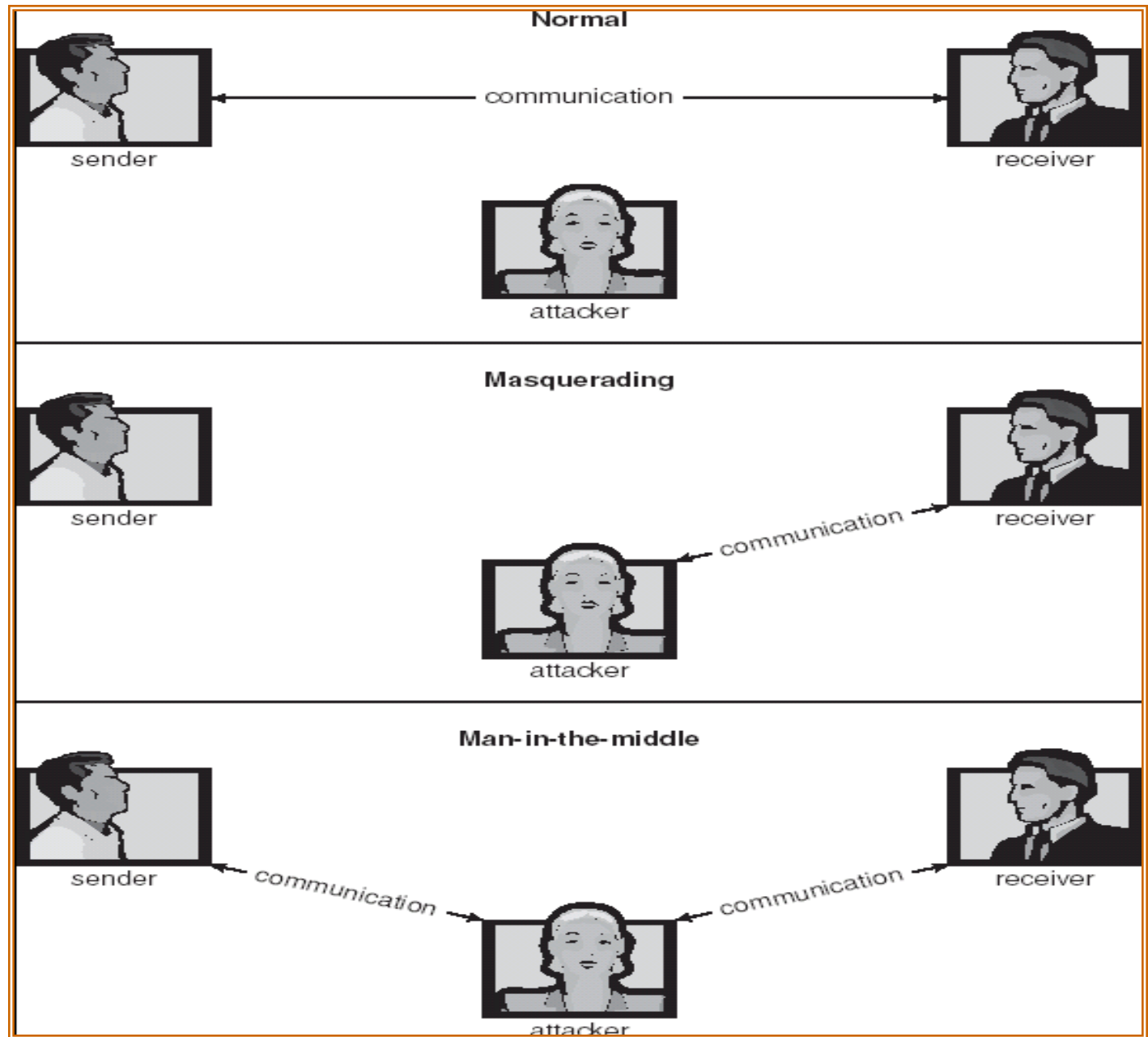
**Spurious messages**      Introducing bogus messages with valid addresses and consistency criteria.

**Site impersonation**      Claiming to be some other logical node.

**Replay**      of previous transmission - repeating previous valid messages. (authorization of cash withdrawal.)

# Typical Security Attacks

## ATTACK METHODS:



# Typical Security Attacks

## ATTACK METHODS:

- Trojan Horse
  - Code segment that misuses its environment
  - Exploits mechanisms for allowing programs written by users to be executed by other users
  - **Spyware, pop-up browser windows, covert channels**
- Trap Door
  - Specific user password that circumvents normal security procedures
  - Could be included in a compiler
- Logic Bomb
  - Program that initiates a security incident under certain circumstances
- Stack and Buffer Overflow
  - Exploits a bug in a program (overflow either the stack or memory buffers)

# Typical Security Attacks

## Viruses

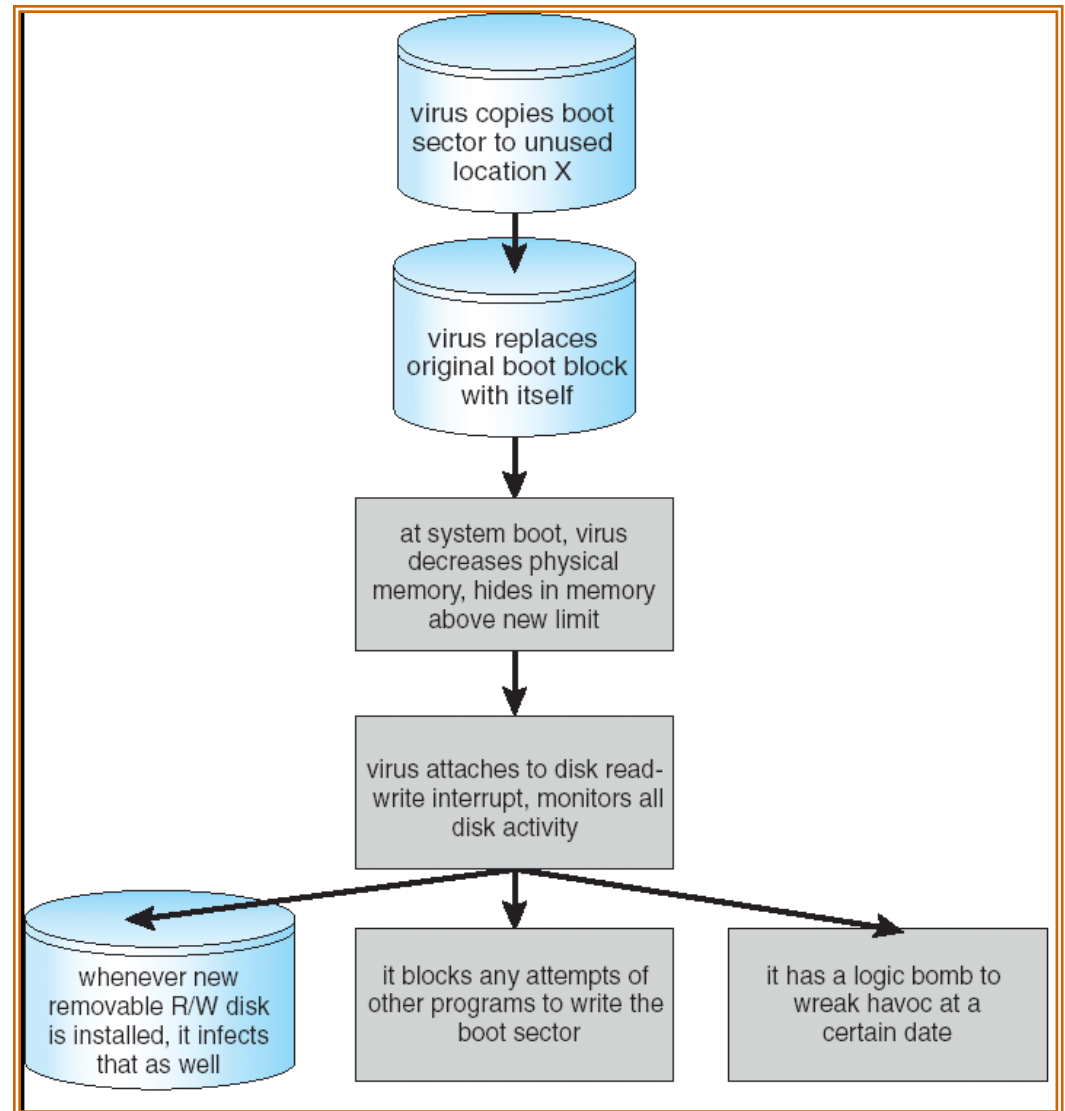
- Code fragment embedded in legitimate program
- Very specific to CPU architecture, operating system, applications
- Usually borne via email or as a macro
  - Visual Basic Macro to reformat hard drive

```
Sub AutoOpen()  
Dim oFS  
Set oFS =  
CreateObject(''Scripting.FileSystemObject'')  
vs = Shell(''c:command.com /k format c:'',vbHide)  
End Sub
```



# Typical Security Attacks

## A Boot Sector Virus



# Typical Security Attacks

## System And Network Threats

- Worms – use **spawn** mechanism; standalone program
- Internet worm
  - Exploited UNIX networking features (remote access) and bugs in *finger* and *sendmail* programs. (See next slide)
  - **Grappling hook** program uploaded main worm program
- Port scanning
  - Automated attempt to connect to a range of ports on one or a range of IP addresses
- Denial of Service
  - Overload the targeted computer preventing it from doing any useful work
  - Distributed denial-of-service (**DDOS**) come from multiple sites at once

# Stuxnet

**Stuxnet** is a computer worm discovered in June 2010. It initially spreads via Microsoft Windows, and targets Siemens industrial software and equipment.

Different variants of Stuxnet targeted five Iranian organizations, with the probable target widely suspected to be the uranium enrichment infrastructure in Iran.

It is initially spread using infected removable drives such as USB flash drives, and then uses other exploits and techniques to infect and update other computers inside private networks that are not directly connected to the Internet.

The malware has both user-mode and kernel-mode rootkit capability under Windows, and its device drivers have been digitally signed with the private keys of two certificates that were stolen from two separate companies. The driver signing helped it install kernel mode rootkit drivers successfully and therefore remain undetected for a relatively long period of time.

Once installed on Windows Stuxnet infects files belonging to Siemens' control software<sup>13</sup> and subverts a communication library. Doing so intercepts communications between software running under Windows and the target Siemens devices. The malware can install itself on PLC devices unnoticed.

Stuxnet malware periodically modifies a control frequency to and thus affects the operation of the connected centrifuge motors by changing their rotational speed.

This causes the centrifuges to be destroyed.

# Authentication

## Password stealing

- Easiest way is through social means

  - fake deposit slips

  - easily guessable passwords

  - calling people on the phone and asking for passwords (or Credit Card numbers, for that matter)

- Technological approaches also

  - simple one: leave program running on a terminal that fakes the login sequence. Capture user name and password to a file and then exit with a fake error message, returning control to the real login process

- Unix password files used to be openly available (encrypted password). Lends itself to brute-force cracking. Unfortunately some programs require access to the password file to run (e.g., mail) → also unfortunately Unix only uses first eight characters of password

SecurID – uses a preprogrammed string of characters

# NSA Exploitation

Edward Snowden made public documents that reveal Government agencies:

- consider it essential to be able to view encrypted data
- have adopted a battery of methods in their assault on this biggest threats

Those methods include

- control over setting of international encryption standards,
- the use of supercomputers to break encryption with "brute force",
- Collaboration with technology companies and internet service providers themselves
- “Man in the middle” attacks on the communication channels themselves.

# SECURITY

## DEFINITIONS:

### Encryption:

$$C = E(M, K_e)$$

E = Encyphering Algorithm

M = Message - plain text

$K_e$  = Encryption key

C = Cyphered text

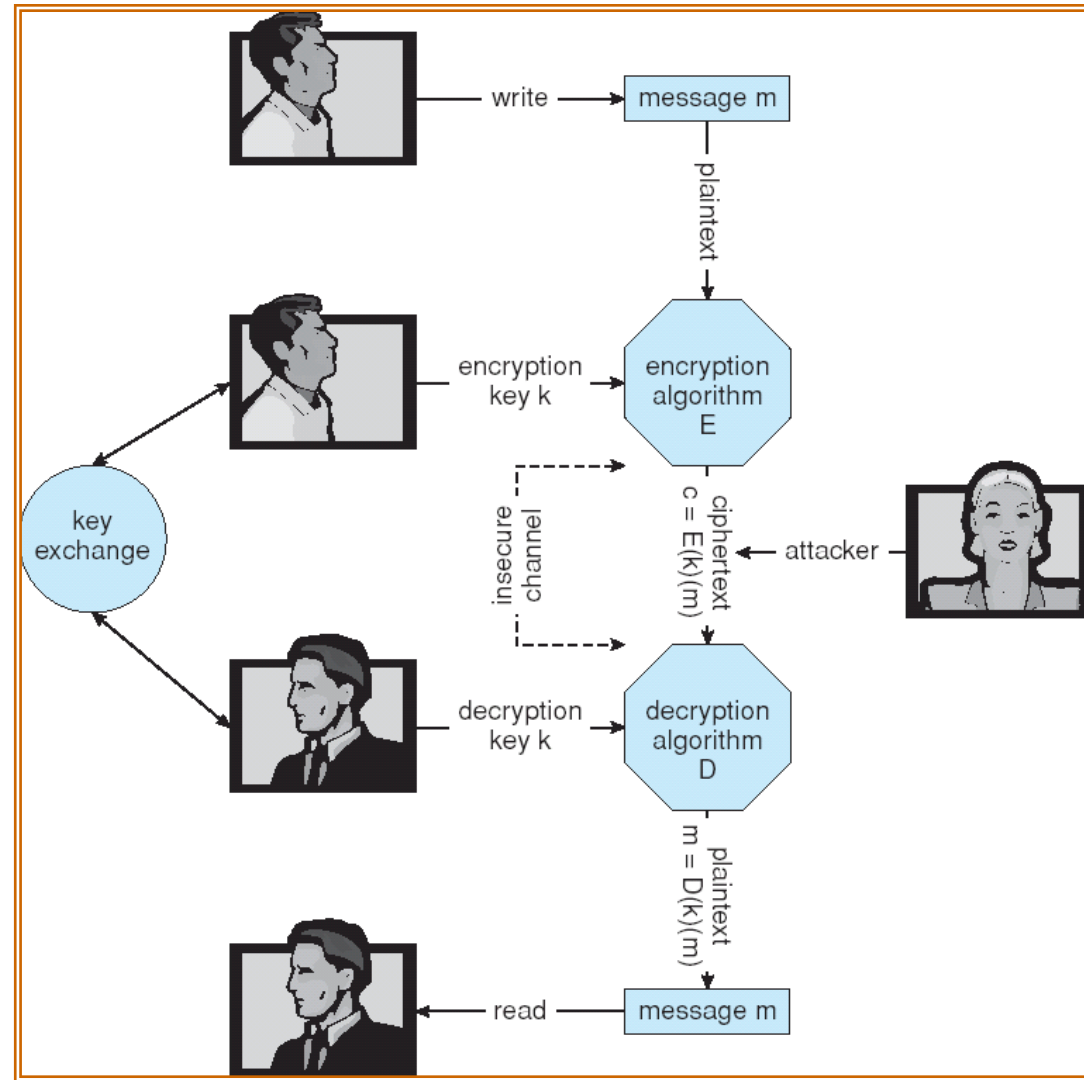
### Decryption:

$$M = D(C, K_d)$$

D = Decyphering Algorithm

$K_d$  = Decryption key

## Cryptography



## DEFINITIONS:

### Cryptosystems are either Conventional or Public Key

- Conventional is symmetric;  $K_e = K_d$  , so the key must be kept secret. Algorithms are simple to describe, but complex in the number of operations.
- Public key is asymmetric;  $K_e \neq K_d$  , so  $K_e$  can be made public.  $K_d$  is secret and can't easily be derived from  $K_e$  .

### Security against attack is either:

- **Unconditionally secure** -  $K_e$  can't be determined regardless of available computational power.
- **Computationally secure:** - calculation of  $K_d$  is economically unfeasible ( it would overwhelm all available computing facilities.)

### The only known unconditionally secure system in common use!

- Involves a random key that has the same length as the plain text to be encrypted.
- The key is used once and then discarded. The key is exclusively OR'd with the message to produce the cypher.
- Given the key and the cypher, the receiver uses the same method to reproduce the message.

# Data Encryption Standard

## DATA ENCRYPTION STANDARD ( DES ):

- The official National Institute of Standards and Technology (NIST), (formerly the National Bureau of Standards) encryption for use by Federal agencies.
- The source of security is the non-linear many-to-one function applied to a block of data. This function uses transposition and substitution. The algorithm is public, but the key (56 bits) is secret.
- Computational power today can crack a 56 bit code.
- In common use today is Triple DES in which 3 different keys are used, making the effective key length 168 bits.

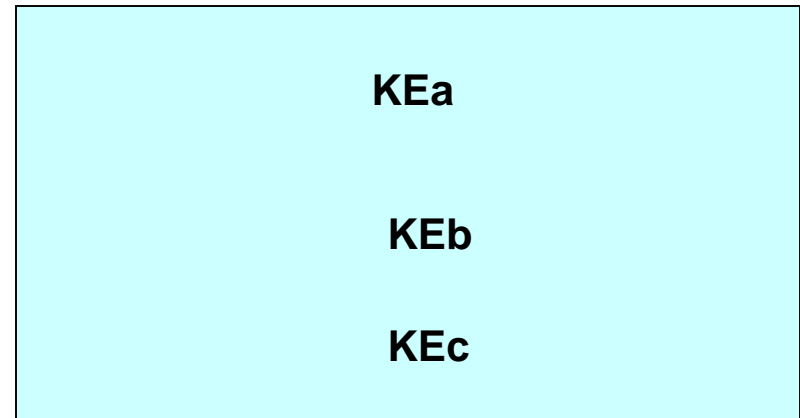


# Public Key Cryptosystems

The general principle is this:

1. Any **RECEIVER A** uses an algorithm to calculate an encryption key **KEa** and a decryption key **KDa**.
2. Then the receiver PUBLICIZES **KEa** to anyone who cares to hear. But the receiver keeps secret the decryption key **KDa**.
3. **User B** sends a message to **A** by first encrypting that message using the publicized key for that receiver **A**, **KEa**.
4. Since only **A** knows how to decrypt the message, it's secure.

Public Key Repository



# Public Key Cryptosystems

To be effective, a system must satisfy the following rules:

- a) Given plaintext and ciphertext, the problem of determining the keys is computationally complex.
- b) It is easy to generate matched pairs of keys  $K_e$ ,  $K_d$  that satisfy the property  
$$D(E(M, K_e), K_d) = M.$$

This implies some sort of trapdoor, such that  $K_e$  and  $K_d$  can be calculated from first principles, but one can't be derived from the other.

- c) The encryption and decryption functions  $E$  and  $D$  are efficient and easy to use.
- d) Given  $K_e$ , the problem of determining  $K_d$  is computationally complex.

What is computationally difficult? Problems that can't easily be calculated in a finite time. Examples include: factoring the product of two very large prime numbers; the knapsack problem.

These problems are NP complete - solution times are exponential in the size of the sample.

# Public Key Cryptosystems

To be effective, a system must satisfy the following rules:

- e) For almost all messages it must be computationally unfeasible to find ciphertext key pairs that will produce the message.

(In other words, an attacker is forced to discover the true (M,Ke) pair that was used to create the ciphertext C.)

- f) Decryption is the inverse of encryption.

$$E( D( M, K_d ), K_e ) = D( E( M, K_e ), K_d )$$

# SECURITY

## Public Key Cryptosystems

### AN EXAMPLE:

1. Two large prime numbers  $p$  and  $q$  are selected using some efficient test for primality. These numbers are secret:

Let  $p = 3$ ,  $q = 11$

2. The product  $n = p * q$  is computed.

$n = 3 * 11 = 33$ .

3. The number  $K_d > \max(p, q)$  is picked at random from the set of integers that are relatively prime to  $L(n) = (p - 1)(q - 1)$ .

$L(n) = (p - 1)(q - 1) = 20$ .  
Choose  $K_d > 11$  and prime to 20.  
Choose  $K_d = 13$ .

4. The integer  $K_e$ ,  $0 < K_e < L(n)$  is computed from  $L(n)$  and  $K_d$  such that  $K_e * K_d = 1 \pmod{L(n)}$ .

$0 < K_e < 20$   
 $K_e = 17$ . (since  $17 * 13 = 221 = 1 \pmod{20}$ )

# Public Key Cryptosystems

## AN EXAMPLE:

Separate the text to be encoded into chunks with values  $0 - (n - 1)$ .

In our example, we'll use  $\langle \text{space} = 0, A = 1, B = 2, C = 3, D = 4, E = 5 \rangle$ .

Then " B A D <sp> B E E " --> "21 04 00 25 05"

$21^{17} \pmod{33} = 21.$	$21^{13} \pmod{33} = 21.$
$04^{17} \pmod{33} = 16.$	$16^{13} \pmod{33} = 04.$
$00^{17} \pmod{33} = 00.$	$00^{13} \pmod{33} = 00.$
$25^{17} \pmod{33} = 31.$	$31^{13} \pmod{33} = 25.$
$05^{17} \pmod{33} = 14.$	$14^{13} \pmod{33} = 05.$

This whole operation works because, though  $n$  and  $K_e$  are known,  $p$  and  $q$  are not public. Thus  $K_d$  is hard to guess.

[Note: recently a 100 digit number was successfully factored into two prime numbers.]

## AUTHENTICATION AND DIGITAL SIGNATURES:

### Sender Authentication:

In a public key system, how does the receiver know who sent a message (since the receiver's encryption key is public)?

Suppose **A** sends message **M** to **B**:

- a) **A** DECRYPTS **M** using **A's**  $K_d(A)$ .
- b) **A** attaches its identification to the message.
- c) **A** ENCRYPTS the entire message using **B's** encryption,  $K_e(B)$   
$$C = E( (A, D(M, K_d(A))), K_e(B) )$$
- d) **B** decrypts using its private key  $K_d(A)$  to produce the pair **A**,  $D(M, K_d(A))$ .
- e) Since the proclaimed sender is **A**, **B** knows to use the public encryption key  $K_e(A)$ .

### Capture/Replay

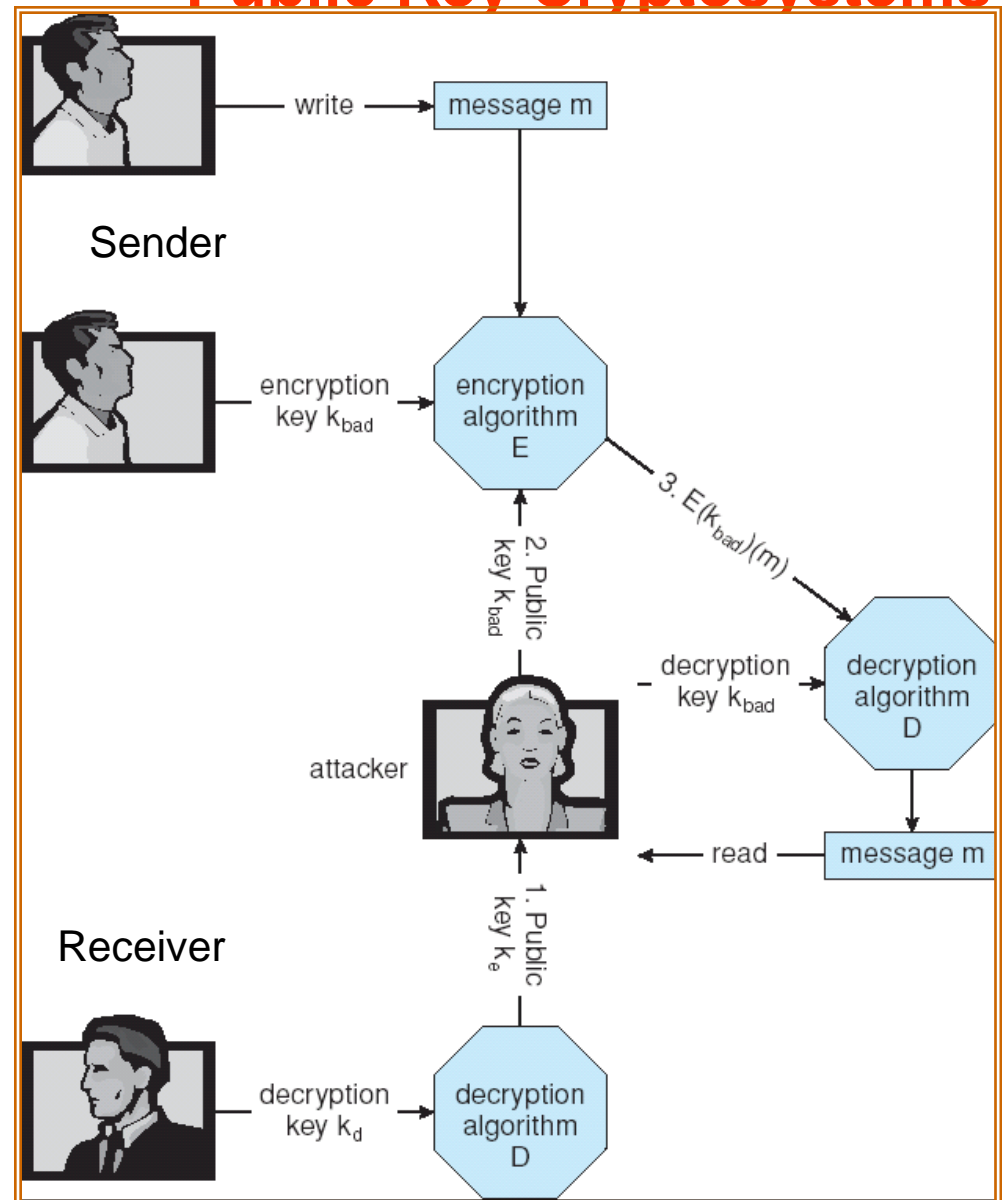
In this case, a third party could capture / replay a message.

The solution is to use a rapidly changing value such as time or a sequence number as part of the message.

## Man-in-the-middle Attack on Asymmetric Cryptography

Here are the attack steps for this scenario:

1. Sender wishes to send a message to Receiver.
2. S asks R for its encryption key.
3. When R returns key, that key is intercepted by the attacker who substitutes her key.
4. Sender encrypts message using this bogus key and returns it.
5. Since the attacker is the owner of this bogus key, the attacker can read the message.



## Example - SSL

- Insertion of cryptography at one layer of the ISO network model (the transport layer)
- SSL – Secure Socket Layer (also called TLS)
- Cryptographic protocol that limits two computers to only exchange messages with each other
  - Very complicated, with many variations
- Used between web servers and browsers for secure communication (credit card numbers)
- The server is verified with a **certificate** assuring client is talking to correct server
- Asymmetric cryptography used to establish a secure **session key** (symmetric encryption) for bulk of communication during session
- Communication between each computer uses symmetric key cryptography



## Example – Windows 7

- Security is based on user accounts
  - Each user has unique security ID
  - Login to ID creates **security access token**
    - Includes security ID for user, for user's groups, and special privileges
    - Every process gets copy of token
    - System checks token to determine if access allowed or denied
- Uses a subject model to ensure access security. A subject tracks and manages permissions for each program that a user runs
- Each object in Windows XP has a security attribute defined by a security descriptor
  - For example, a file has a security descriptor that indicates the access permissions for all users

# Security Classifications

U.S. Department of Defense outlines four divisions of computer security: **A**, **B**, **C**, and **D**.

- **D** – Minimal security.
- **C** – Provides discretionary protection through auditing. Divided into **C1** and **C2**. **C1** identifies cooperating users with the same level of protection. **C2** allows user-level access control.
- **B** – All the properties of **C**, however each object may have unique sensitivity labels. Divided into **B1**, **B2**, and **B3**.
- **A** – Uses formal design and verification techniques to ensure security.

# SECURITY

## Wrap Up

In this chapter we've looked at how to secure information that may be placed in hazardous public forums.

Data on the net is an excellent example here.